IN THE SPECIFICATION

The specification as amended below with replacement paragraphs shows added text with underlining and deleted text with ~~strikethrough~~.

Please REPLACE the paragraph numbered [0006] at page 4, with the following paragraph:

[0006] In the cache system in a multiprocessor system, cache coherence is managed such that inconsistency between a cache device in one processor and a cache device in another processor is not caused. For example, as shown in FIG. 3A, data stored in a plurality of cache devices 100-1 to 100-n are shared. In the case that the processor 100-n performs ~~wiring~~ writing on the shared data, the writing is performed after the processor 100-n informs the other cache devices 100-1 and 102 having the same data that the writing will be performed to make the present data in the cache devices invalid without fail, as shown in FIG. 3B. By the invalidation, the other cache devices can know that the data that they have are not newest. The method that all processors can read the newest data at the time of the reading in this manner is cache coherency management. In the pre-fetch, one cache device predicts data that will be requested before long and reads the data as well as data required by the processor. However, this prediction does not necessarily prove to be right. Thus, useless data may be read. Even in the case of a single processor system, useless reading by the pre-fetch causes a problem, such as useless traffic between the main memory and a cache. In the case of a multiprocessor system, not only the useless traffic but also useless sharing arises. In other words, data that is not shared in the methods of reading only required data may be shared by plural caches in the methods of using the pre-fetch. It is necessary that at the time of writing onto the shared data in the cache device, the cache device informs the other cache devices that the writing will be performed. As far as this processing of informing the other cache devices is not finished, any data cannot be renewed. Therefore, the writing on the shared data is heavy processing, that is, processing which requires much time in the cache device. As a result, in the pre-fetch in any multiprocessor system, the drawbacks of the useless traffic and the useless sharing cancel the advantage of the pre-fetch. Thus, the multiprocessor system does not exhibit superior performance. As described above, in conventional cache devices, pre-fetch, which improves the ratio of hits on the caches, also results in an increase in overhead at the time of writing due

to useless sharing and an increase in data-transmission by the pre-fetch. As a result, a problem that the pre-fetch is not easily applied to multiprocessor systems arises.

Please REPLACE the paragraph numbered [0020] at page 13, with the following paragraph:

[0020] It is difficult to apply the pre-fetch protocol of the present invention to normal data. Specifically, a normal data read request is generated when data becomes necessary in the processor. The fact that the reading of the data ends in failure and then the data is not obtained results in a drop in the operation performance of the ~~profesor~~ processor. Therefore, the first weak read, the second weak read, and the passive preservation mode (P) of the present invention are applied only to reading of speculative data such as pre-fetch. In other words, normal data is stored in the normal preservation mode (N) in the cache device. Pre-fetch data, which is passive preservation data, is stored in the passive preservation mode (P). They are distinguished in the cache memory. In this manner, the cache device wherein preservation modes are set in the respective cache lines can be made.

Please REPLACE the paragraph numbered [0056] at page 31, with the following paragraph:

[0056] FIG. 10 is a flowchart of pre-fetch processing by the first weak protocol read processing unit shown in FIG. 7. In the case that pre-fetch of data in the requested addresses ADR + n is requested by the hardware, which accompanies a memory request from the processor 18, in step S1 the processor 18 first searches its own cache memory 32 and then in step S2 the processor 12 judges whether or not the corresponding data are present in the cache and are hit. When the data are hit, in step S9 read data are given back to the processor and then the successive processing is finished. When the corresponding data are not present in the cache and are not hit in step S2, in step S3 a request message for read based on the weak protocol is sent into the common bus 14. In response to the request message for read based on the weak protocol, in each of the other cache devices it is examined whether the data corresponding to the requested addresses ADR + n are stored and, if stored, the state of the data is examined. As a result, the corresponding state controlling line is asserted. In step S4, it is first checked whether or not the state is the exclusive state (E) based on assertion of the EX line or the data-

modified state (M) based on assertion of the HITM line. If the state controlling line corresponding to the exclusive state (E) and the data-modified state (M) is asserted, the present processing goes to step S5 so that the reading of the pre-fetch-requested date is regarded as failure and the processing is finished. If the EX line or the HITM line is not asserted in step S4, the processing goes to step S6 so that it is checked whether or not the present state is the data-shared state (S) based on assertion of the ~~HITM~~ HIT line. If the ~~HITM~~ HIT line is asserted, the present state is the data-shared state (S). Therefore, the processing goes to step S7, so that data in the requested addresses ADR + n are read and transmitted from the main memory 18 and then the data are made into the data-shared state (S) and stored in the cache array. If the ~~HITM~~ HIT line is not asserted in step S6, the present state is the invalid state (I). Thus, the present processing advances to the step S8, so that data in the requested addresses ADR + n are read and transmitted from the main memory 18 in the same manner and then the data are stored in the exclusive state (E) in the cache array. As described above, in the case that it is necessary to modify the other cache devices by pre-fetch, that is, in the case that the data corresponding to pre-fetch are being stored in the exclusive state (E) or the data-modified state (M) in the other cache devices, the pre-fetch request is made to a read operation based on the weak protocol and the reading of the pre-fetch-requested data is regarded as failure to interrupt the processing. Thus, when the processor accesses the data in the exclusive state (E) or the data-modified state (M) in some other cache devices, in the cache devices in which the same data is stored as pre-fetch data an operation for making their data state into the data-shared state (S) becomes unnecessary. It is also possible to reduce overhead at the time of writing by the processor in the cache device in which data is stored in the exclusive state (E) or the data-modified state (M).

Please REPLACE the paragraph numbered [0064] at page 43, with the following paragraph:

[0064] In the case that data is stored in the (P) mode in any one of the cache memories, the cache controller 30 for causing the data to be stored in the (P) mode carries out the following operation through the function of the passive reading mode processing unit 48 shown in FIG. 7 when a request message based on the normal read or the weak protocol read is sent out into the common bus by a memory access request from some other cache devices. First, the state-examination-result based on the request message sent into the common bus 14 is waited. In

this case, the cache controller 30 in which the data in the (P) mode is stored does not assert any of the EX line, the HIT line and the HITM line regardless of the state of the cache so that to the other cache controllers appear the invalid state (I) when they are observed from the outside. In the case that any of the Invalid line, EX line, ~~the HIT line~~ and the HITM line is not asserted on the basis of the result of state-examination in the other cache devices, that is, in the case that the cache controllers other than the cache controller for causing the data in the (P) mode to be stored are invalid (I), this controller 30 changes the state of the data in the (P) mode to the invalid state (I) in order to invalidate its own data. In this manner, the other cache controller which issued the read request can read data in the requested address, in the exclusive state (E), from the main memory 18 and cause the data to be stored. In the case that the Ex line or the HITM line is asserted and the data is stored, in the exclusive state (E) or the data-modified state (M), in any one of the other cache devices, the cache controller 30 for causing the data to be stored in the (P) mode changes the state of the data in the (P) mode to the invalid state (I). On the other hand, in the case that the other cache controllers are in the data-shared state (S), wherein the HIT line is asserted, on the basis of the result of the state-examination of the other controllers the cache controller 30 for causing the data in the (P) mode to be stored does not need to change its own data.